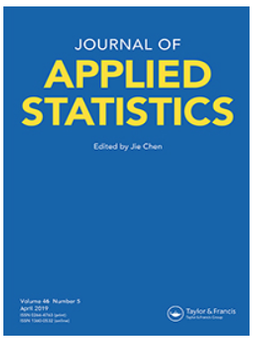

Journal of Applied Statistics

ISSN: 0266-4763 (Print) 1360-0532 (Online) Journal homepage: https://www.tandfonline.com/loi/cjas20

Modeling association between multivariate correlated outcomes and high-dimensional sparse covariates: the adaptive SVS method

J. Pecanka, A. W. van der Vaart & M. A. Jonker

To cite this article: J. Pecanka, A. W. van der Vaart & M. A. Jonker (2019) Modeling association between multivariate correlated outcomes and high-dimensional sparse covariates: the adaptive SVS method, Journal of Applied Statistics, 46:5, 893-913, DOI: 10.1080/02664763.2018.1523377

To link to this article: https://doi.org/10.1080/02664763.2018.1523377

Published online: 19 Sep 2018.

JOURNAL OF APPLIED STATISTICS
2019, VOL. 46, NO. 5, 893–913
https://doi.org/10.1080/02664763.2018.1523377

Taylor & Francis
Taylor & Francis Group

Modeling association between multivariate correlated outcomes and high-dimensional sparse covariates: the adaptive SVS method

J. Pecanka [a], A. W. van der Vaart [b] and M. A. Jonker [c]

[a]Department of Medical Statistics and Bioinformatics, Leiden University Medical Center, Leiden, Netherlands; [b]Mathematical Institute, Faculty of Science, Leiden University, Leiden, Netherlands; [c]Department for Health Evidence – Biostatistics, Radboud University Medical Center, Nijmegen, Netherlands

ABSTRACT

The problem of modeling the relationship between a set of covariates and a multivariate response with correlated components often arises in many areas of research such as genetics, psychometrics, signal processing. In the linear regression framework, such task can be addressed using a number of existing methods. In the high-dimensional sparse setting, most of these methods rely on the idea of penalization in order to efficiently estimate the regression matrix. Examples of such methods include the lasso, the group lasso, the adaptive group lasso or the simultaneous variable selection (SVS) method. Crucially, a suitably chosen penalty also allows for an efficient exploitation of the correlation structure within the multivariate response. In this paper we introduce a novel variant of such method called the *adaptive SVS*, which is closely linked with the adaptive group lasso. Via a simulation study we investigate its performance in the high-dimensional sparse regression setting. We provide a comparison with a number of other popular methods under different scenarios and show that the adaptive SVS is a powerful tool for efficient recovery of signal in such setting. The methods are applied to genetic data.

ARTICLE HISTORY
Received 23 March 2017
Accepted 7 September 2018

KEYWORDS
Multivariate outcome; high-dimensional regressors; penalty; simultaneous variable selection; lasso; alternative splicing

1. Introduction

In this paper we focus on the problem of modeling and discovering association between genotypes, or in principal any covariates of interest, and multivariate correlated phenotypes. We focus on exploratory analysis based on regression, where genetic loci are used as explanatory variables for a multivariate phenotype, which is the dependent variable. Although we primarily formulate the problem and the methods within a genetic setting where the number of covariates of interest is larger than the dimension of the response, the methodology can be readily applied in many other areas with similar characteristics. Our results were obtained for a general setting in which the response (e.g. phenotype) is assumed to be generated via a factor model from the underlying covariates (e.g. genetic

CONTACT M. A. Jonker marianne.jonker@radboudumc.nl

variants). In a genetic setting, the phenotype typically consists of several related quantitative measurements which together describe the condition of interest, while the goal is to find a link between the condition and a subset of the available genetic variants. Since the observed quantitative measurements all relate to a single underlying medical condition, they often exhibit correlation (dependence). If considered separately, these measurements are typically only loosely associated with the patient's genotype, which makes identifying the association a very difficult task. Therefore, they should be treated simultaneously in a way that reflects this correlation in order to increase the statistical efficiency of the analysis. In [1] a methodology is described for finding a linear combination of the multiple phenotypes that maximizes the evidence for association with a SNP (single nucleotide polymorphism). The paper [8] gives an overview of many existing methods for genetic association studies with multiple outcomes, and compares a number of methods that are directly available in genetics software. In the present paper we restrict ourselves to methods based on a multiple-response multivariate regression.

Within the genetic context the multiple-response multivariate linear regression approach must cope with the usual complication of high dimensionality of the regressors: the sample size (the number of individuals) is smaller than the number of available regressors. Typically, this high-dimensionality problem is dealt with by using penalized regression methods, which have been extensively studies for the last two decades and much knowledge has been accumulated both in the classical and especially in the high-dimensional setting by Bickel *et al.* [2], Bühlmann *et al.* [4], Castillo *et al.* [5], Castillo and van der Vaart [6], Zhang and Huang [30] and others. However, only some of these methods properly reflect the nature of the current problem of modeling multivariate phenotypes with correlated coordinates. Available methods include the *graph-guided fused lasso* (GFlasso) [12] and *simultaneous variable selection* (SVS) [23,24].

In this paper we propose an adaptation of the SVS method by changing the estimation objective function so that it better reflects the information within the data to which it is applied. This adaptive SVS method is directly linked to two existing statistical methods. On the one hand, it is an extension of the SVS method [24]. On the other hand, it is closely linked with the adaptive group lasso method [25]. Extensive simulation studies are performed to compare its performance with other penalty methods that are often used for modeling data with multiple-response and high-dimensional multivariate covariates.

We describe the SVS and the proposed adapted SVS method in Section 2. Next, we investigate the performance of the adaptive SVS method and compare it with several other suitable methods by means of several simulation studies in Section 3. Finally, in Section 4 we apply the adaptive SVS method to an eQTL analysis of an existing expression data set, where we look for SNP-driven gene expression regulation.

The content of this paper is based on Part III of the first author's PhD thesis [18], for which the other two authors served as supervisors.

2. Methods

2.1. Assumptions and notation

We assume an $n \times p$-dimensional random matrix of responses $\boldsymbol{Y} = (Y_{ij})$ with independent rows and (possibly) dependent columns. Further, we assume we have an

$n \times q$-dimensional regression matrix $\boldsymbol{X} = (x_{ij})$ and a $q \times p$-dimensional (non-random) matrix of regression parameters $\boldsymbol{B} = (\beta_{ij})$ such that

$$\boldsymbol{Y} = \boldsymbol{XB} + \boldsymbol{E}, \tag{1}$$

where $\boldsymbol{E}$ is an $n \times p$-dimensional random matrix with zero mean and independent rows and variance matrix $\boldsymbol{\Sigma}$. To eliminate the need for intercept in model (1), we center (on a per-column basis) both the response matrix $\boldsymbol{Y}$ and the regression matrix $\boldsymbol{X}$. In the intended genetic application the matrix $\boldsymbol{X}$ contains genetic information at q loci and the rows of $\boldsymbol{Y}$ can be seen as a p-dimensional multivariate phenotype.

Since the number of parameters in the full model far exceeds the number of available observations, a reliable inference about the model's parameters is a daunting task which demands additional assumptions about the modeled phenomena. A popular approach to dealing with such systems is the concept of *sparsity*, which is the notion that the observed response is in fact influenced only by a subset of the available explanatory variables; there exists a combination of $i \in \{1, \ldots, q\}$, $j \in \{1, \ldots, p\}$ such that $\beta_{ij} = 0$. Furthermore, the assumption of *common association* is often made in this setting: For any $i \in \{1, \ldots, q\}$, $j \in \{1, \ldots, p\}$ it holds $\beta_{ij} = 0$ if and only if $\beta_{ik} = 0$ for all $k \neq j$. This means that if one of the components of the multivariate phenotype is associated with a locus, then all other components of the phenotype are also associated with that locus, and vice-versa. For discussion of this assumption, see for instance the discussions to [1].

2.2. Simultaneous variable selection

The new method proposed in this paper is a generalized version of the SVS method, which is described in this subsection. The SVS penalized method was specifically designed for the multiple-response multivariate regression model. It is based on the idea to penalize the sum of squared residuals by the sum of ℓ_α norms of the rows of the parameter matrix $\boldsymbol{B}$, thereby forcing the resulting estimates closer together. The SVS estimator for $\boldsymbol{B}$ is defined by

$$\widehat{\boldsymbol{B}}_{\mathrm{SVS}}(\lambda) = \arg\min_{\boldsymbol{B}} \|\boldsymbol{Y} - \boldsymbol{XB}\|_2^2 + \lambda \sum_{k=1}^{q} \|\boldsymbol{\beta}_{k\bullet}\|_\alpha, \tag{2}$$

where $\lambda > 0$ is a tuning constant and $\boldsymbol{\beta}_{k\bullet}$ is the kth row of the matrix $\boldsymbol{B}$ and α is a positive constant determining the type of norm that is applied to the rows of the parameter matrix $\boldsymbol{B}$ before these norms are summed up to create a penalty. The penalty is a combined ℓ_1/ℓ_α type norm; the ℓ_1 part is to achieve sparsity among regressors (only a handful of active/selected SNPs), while the ℓ_α norm over the vector of effects of an individual regressor (a SNP) on the various components of the phenotype leads to shrinkage to zero of the group of coefficients for that regressor. The value of α determines the specific degree and type of shrinkage. In other words, the common association assumption is reflected in the penalty primarily by the ℓ_α norm at SNP level (for $\alpha > 1$). Shrinkage to zero due to the ℓ_α norm is easiest to understand for $\alpha = 2$, when the penalty corresponds to ridge regression, a method that is well known to shrink parameter estimates simultaneously to zero. For larger values of α, and for $\alpha = \infty$ in particular, the penalty becomes increasingly more sensitive to the value of the largest effect of a regressor towards the phenotype. The case

with $\alpha = 1$ turns SVS into the lasso, at which point the grouping effect is absent. In this case there is no qualitative difference in treatment by the penalty of effects corresponding to the same and to different SNPs. Consequently, the penalized method does not conform to the common association assumption. See, for example, Hastie *et al.* [9] for discussion of the shrinkage effects of the various penalties.

The case $\alpha = \infty$ is referred to as the $L_\infty-$ SVS [16,24], and the case of $\alpha = 2$ as L_2- SVS [15,17,20,24]. The L_2- SVS can be viewed as a special case of the group lasso estimator with q groups, where each of the groups corresponds to one coordinate of the multivariate phenotypes, and the weight matrices are all equal to the p-identity matrix I_p [29,32].

2.3. Adaptive simultaneous variable selection

We define the *adaptive SVS* (aSVS) estimator of the parameter matrix $\boldsymbol{B}$ of (1) as

$$\widehat{\boldsymbol{B}}_{\text{aSVS}}(\lambda) = \arg\min_{\boldsymbol{B}} \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}\|_2^2 + \lambda \sum_{k=1}^{q} \pi_k \|\boldsymbol{\beta}_{k\bullet}\|_\alpha, \tag{3}$$

where $\lambda > 0$ is a tuning parameter and π_k, $k = 1, \ldots, q$, are non-negative *penalization weights*. In the paper we assume that the penalization weights are scaled so that they satisfy $\sum_k \pi_k = q$. This results in no loss of generality provided that all π_k are finite. The adaptive SVS with $\alpha = \infty$, $\alpha = 1$ and $\alpha = 2$ are referred to as $L_\infty-$ aSVS, L_1-aSVS and L_2- aSVS, respectively. Putting $\alpha = 1$ leads to a version of the *adaptive lasso* [31], while $\alpha = 2$ is also known as the adaptive group lasso [25].

The motivation behind the penalization weights is to change the degree by which each regressor (SNP) contributes to the value of the penalty, thus making the estimate *adapt* to additional information available to the user as expressed through π_k. A large value of π_k means that the kth regressor (SNP) is heavily penalized relative to the regressors with small penalization weights, thus restricting the freedom of the kth regressor parameter estimate and making it deflated compared to their (non-adaptive) SVS counterparts. On the other hand, a small value of π_k gives more freedom to the estimates for the kth regressor thus allowing them to inflate, while $\pi_k = 0$ leads to no penalization for the kth regressor. Thus the presence of weights may lead to an additional grouping mechanism among the estimated effects of an individual regressor.

As we show below, applying this simple modification to the form of the penalty can significantly improve the performance of the SVS method. In order to achieve this, however, the penalization weights π_k in Equation (3) must be suitably selected. One way to do this is to employ an initial data-based estimator of $\boldsymbol{B}$ and use it to define π_k and let the method *adapt* to the initial estimate of $\boldsymbol{B}$, hence the name *adaptive SVS*. If the initial estimator is reasonable enough, and is used sensibly, the adaptation should result in an efficiently performing method. For instance, with $\alpha = 1$ the adaptive SVS turns into the adaptive lasso and its consistency and oracle properties immediately translate to the adaptive SVS with $\alpha = 1$ [10]. Moreover, for $\alpha = 2$ the adaptive SVS turns into the adaptive group lasso and its consistency and oracle properties translate to the adaptive SVS with $\alpha = 2$ [27,28]. Consequently, it appears reasonable to expect that other adaptive variants of the

adaptive SVS might also perform well under similar conditions. For other optimality conditions and a more general treatment of the theoretical properties of these methods we refer to [3].

2.3.1. Choice of penalization weights

For good performance the penalization weights must be chosen suitably, given the clear danger that regression coefficients of truly associated covariates might be shrunk too close to zero, and conversely for the non-associated covariates. We investigate using penalization weights based on ordinary least squares (OLS) estimates resulting from univariate regression of each component of the phenotype on each regressor.

In the univariate OLS approach we regress each component of the phenotype on each regressor separately and apply a chosen function to obtain a single weighing factor for each regressor. This effectively boils down to using the Pearson correlation coefficient of each regressor with each component of the phenotype. Such approach is quite advantageous because of its computational simplicity, which contributes to fast analysis. In formulas, we compute $\widehat{\boldsymbol{B}}_{\text{ols}} = (b_{kl}^{\text{ols}})$ according to

$$b_{kl}^{\text{ols}} = \underset{b\in\mathbb{R}}{\arg\min} \|\boldsymbol{Y}_{\bullet l} - b\boldsymbol{X}_{\bullet k}\|^2, \quad k = 1,\ldots,q, \quad l = 1,\ldots,p,$$

and use it as the initial estimate for $\boldsymbol{B}$. Then, since weighing in the adaptive SVS of (3) is done per regressor, for each k we need to transform $b_{kl}^{\text{ols}}, l = 1,\ldots,p$, into a single value π_k. It is desirable that the penalization weight π_k is 'large' whenever the vector b_{kl}^{ols} is 'small', i.e. near zero. One way to achieve this is to first reduce the latter vector (in absolute value) to its mean, and next define the weight as a decreasing function of this mean. Later in the paper we shall study the choices $\pi_k \propto ((1/p)\sum_l |b_{kl}^{\text{ols}}|)^{-\nu}$, for $\nu = 0.5, 1, 2$. In general, the higher the value of ν, the more the adaptive SVS estimator relies on the information carried by the mean univariate effects as to the relative importance of each regressor. Therefore, of the three, the choice $\nu = 0.5$ is the more conservative one and should be a reasonable choice in those applications where the data is expected to be noisy and/or if outliers are suspected. For clean data, on the other hand, the choices of $\nu = 1$ or $\nu = 2$ are likely to yield superior performances, as evidenced by the simulation study below.

In our applications, we must keep in mind that using univariate regression as a proxy for a multivariate model can be tricky unless the design matrix is (near) orthogonal, since with highly correlated regressors many estimates might end up large while in reality only a handful of the corresponding parameters are in fact non-negligible. For such data, these concerns can be mitigated via the choice of ν, or through the use of a different functional form for the weights, such as replacing the mean by the median when determining π_k. However, in the adaptive SVS method the univariate OLS estimates are used only mainly as a basis for the subsequent estimation by the adaptive SVS, which limits the severity of this caveat.

2.3.2. Naive OLS approach

Despite the lack of orthogonality in many applications, which complicates the use of OLS in such setting, the univariate OLS-based analysis is surprisingly popular in many applications such as genetics and psychometrics. Therefore, we include this method in our simulation study, where it is referred to as the *naive OLS* method. Since the method always

yields non-zero estimates, unlike many penalized regression methods, it does not possess an inherent selection property. The usual way to solve this issue is to look at the p-values associated with each univariate estimate and base the selection on them. Given the large number of these p-values a multiple testing correction is necessary in order to avoid a potentially huge number of false selections. To that end we focus on the Bonferroni corrected p-values associated with the univariate OLS estimates.

3. Simulation study

We investigate the performance of the adaptive SVS method on data sets generated from real genetic data. We use the multivariate multiple regression model to generate correlated multivariate phenotypes and we deploy the adaptive SVS and several other methods and compare their relative performances using various measures. Of course, we prevented a possibly spurious advantage of 'using the data twice', once for the choice of weight and once for parameter estimation, by incorporating the determination of the weight into each simulation run. The cross-validation (CV) steps, which were utilized to determine the value of the smoothing parameter λ (as well as the additional parameters in the fusion methods), were similarly accounted for.

3.1. Data simulation

3.1.1. Genotype data

The simulation was based on real life genotype data which contained a total 3000 SNPs for 2000 individuals. The genotypes for each locus were numerically represented as the number of minor alleles at that locus, i.e. each genotype was a sequence of 2000 values from the set $\{0, 1, 2\}$. Figure 1 shows the linkage disequilibrium (LD) heat plot for the full data set. We chose to work with an empirical data set, rather than a simulated set, in view of the difficulty of representing and simulating from a realistic very high-dimensional distribution.

Next we chose 8 of the available 3000 SNPs at random (from the SNPs with MAF at least 0.2) to serve as the *neighbors* of 8 additional *simulated* SNPs, which were subsequently used as causal (i.e. have non-zero regression coefficients). Simulating these additional SNPs

Figure 1. Heat map of linkage disquilibrium patterns as measured by squared correlation coefficient within the real life genetic data set with $q_0 = 3000$ SNPs and $n = 2000$ individuals used as a basis for the simulation study in this paper. SNPs are numbered $1, 2, \ldots, 3000$ from left to right and the empirical SNPs used as neighbors in simulation are selected from SNPs $2001, \ldots, 3000$.

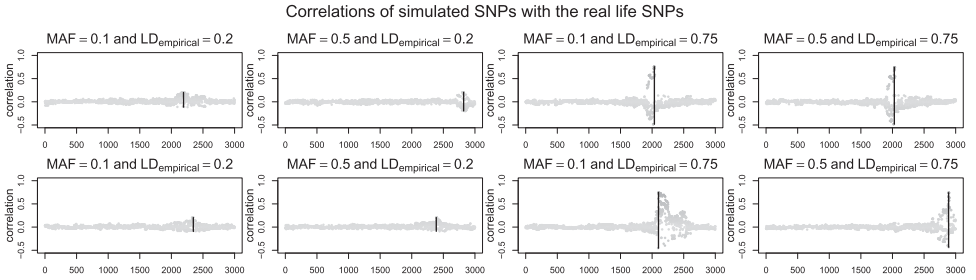

Figure 2. Typical observed sample correlation coefficients between a set of 8 simulated SNPs and the rest of the real life 3000 SNPs. Positions of the simulated SNPs (next to their empirical neighbor SNPs) are denoted by vertical lines, where the heights of these lines express the observed maximum and minimum correlations.

allowed us to control the LD with their neighbors, and also their minor allele frequency. The 8 SNPs were simulated for each individual such that the LD with their neighbor was either 0.2 or 0.75, and the MAFs were either 0.1 or 0.5. We merged the simulated SNP genotypes with the real-life SNP genotypes to obtain a data set representing 3008 SNPs for 2000 individuals, from which we would next construct the design matrix for the further simulations.

Repeating this process of selecting SNPs and simulating genotypes as their neighbors 25 times, we obtained a total of 25 design matrices. For illustration Figure 2 shows the observed correlations between the simulated SNPs and the rest of the full genotype data set for a single, randomly chosen design matrix (out of the collection of 25).

3.1.2. Phenotype data

Starting with $\boldsymbol{X} = (x_{ij})$ we repeatedly simulated p-dimensional phenotypes for each of the n individuals using the linear regression model (1) with $q \times p$ regression matrix $\boldsymbol{B}$. For the simulation of phenotypes we considered several different scenarios (settings of parameters) in order to investigate different aspects of the performances of the considered methods. For given choices of n and p we start by generating a *factor variable* for each of the n individuals through a univariate-response multivariate linear regression model where only the simulated SNP genotypes have non-zero coefficients, henceforth referred to as the *causal* SNPs. Denoting by Q the set of all SNPs (i.e. columns of $\boldsymbol{X} = (x_{ij})$), we simulated a vector of n factor variables $\boldsymbol{F} = (F_1, \ldots, F_n)'$ according to

$$F_i = \sum_{k \in Q} \alpha_k \, x_{ik} + e_i, \quad i = 1, \ldots, n, \tag{4}$$

where $\alpha_k, k \in Q$ were selected in several different ways specified below (see scenarios A, B, C) and e_i are independent zero-mean normally distributed errors with fixed variance 0.2.

The observed factor variables $F_1, \ldots, F_n$ were entered into p univariate linear regression models with normally distributed independent random errors and regression coefficients ranging over predefined set of values, which produced an $n \times p$ matrix of row-wise independent and column-wise correlated responses $\boldsymbol{Y} = (Y_{ij})$. Written in formula the model is

$$Y_{ij} = \gamma_j F_i + e_{ij}, \quad i = 1, \ldots, n, \quad j = 1, \ldots, p. \tag{5}$$

Table 1. Summary of settings used during phenotype simulation via (4) and (5) under scenarios A, B and C.

scenario	A	B	C
phenotypes	$p = 5$	$p = 10$	
sample size	$n = 2000$	$n = 100, 200, \ldots, 2000$	
number of SNPs	$q = 3008$	$q = 1004$	
non-causal SNPs	$q_0 = 3000$	$q_0 = 1000$	
causal SNPs	$q_1 = 8$	$q_1 = 4$	
α_k	$\alpha_k = 0$ for $k = 1, \ldots, 3000$ $\alpha_k = 0.1$ for $k = 3001, \ldots, 3004$ $\alpha_k = 0.15$ for $k = 3005, \ldots, 3008$	$\alpha_k = 0$ for $k = 1, \ldots, 1000$ $\alpha_k = 0.05$ for $k = 1001$ $\alpha_k = 0.1$ for $k = 1002$ $\alpha_k = 0.15$ for $k = 1003$ $\alpha_k = 0.2$ for $k = 1004$	
γ_j	$\gamma_j = j/10$ for $j = 1, \ldots, 5$	$\gamma_j = j/10$ for $j = 1, \ldots, 10$	$\gamma_j = j/10 - 0.5$ for $j = 1, \ldots, 10$
error variance	$\sigma^2 = 0.2$	$\sigma^2 = 0.2$	

Note that Equation (5) implies $\mathbb{E}Y_{ij} = \sum_{k \in Q} \beta_{kj} x_{ik}, i = 1, \ldots, n, j = 1, \ldots, p$, where $\beta_{kj} = \alpha_k \gamma_j$. The idea in Equation (4) is to set only a small fraction of values of $\alpha_k, k \in Q$ non-zero. Respectively, denote the subsets of zero and non-zero coefficients by Q_0 and Q_1 and their sizes by q_0 and q_1. In the three scenarios below, we have $q = q_0 + q_1 = 3008$ and $q_1 = 8$ (scenario A) and $q = 1004$ and $q_1 = 4$ (scenarios B and C) with only the simulated SNPs in Q_1.

Simulating data according to the scheme described by (4) and (5) is designed to induce dependence between the components of the multivariate phenotypes in a way that adheres to the assumptions of *sparsity* and *common association*. We simulated data under the three different scenarios A, B and C. The settings for these scenarios are given in Table 1.

Under scenarios B and C we allowed the sample sizes to vary, which is intended to yield insight into the dynamics of performance by the compared methods in terms of sample size. The way the two scenarios B and C differ is in the choice of the values of the loading parameters $\gamma_1, \ldots, \gamma_{10}$. Under scenario B we put $\gamma_j = j/10$ where $j = 1, \ldots, 10$, which means that all of the loadings take non-zero values and all of them are positive, which should favor the use of naive OLS approach with the summed-up phenotypes. On the other hand, the loadings $\gamma_j = j/10 - 0.5$, for $j = 1, \ldots, 10$, of scenario C lead to a different direction of dependence between the first 4 and 5 last phenotypes (negative and positive, respectively), with the 5th phenotype being noise only. This setup emulates a situation in which the type of relationship between the genotypes and the considered phenotypes is not favorable for the naive OLS with the summed-up phenotypes. It also allowed us to investigate the performance of the methods under slight deviation from the *common association* assumption, which is violated due to γ_5 being zero under scenario C. With the varying signs of the resulting univariate regression coefficients this scenario is also less favorable from the perspective of the specific way in which we determine the weights by averaging the estimates of the univariate regression coefficients.

3.2. Used estimation methods

In the analysis we considered both versions of the adaptive SVS method, namely $L_\infty-$aSVS and L_2-aSVS, each in several different variants based on different choices for the adaptation weights. As performance benchmarks for the adaptive SVS we use the non-adaptive SVS of (2), the naive OLS (with Bonferroni correction), the lasso [22], the adaptive lasso

[31], and the GFlasso of (A.2) (the latter two are described in the appendix of the paper). In order to eliminate the need for an intercept in the models we centered (per-column) all of the phenotype and genotype matrices.

Regarding the GFlasso we used several values for the cut-off parameter ρ, namely 0.05, 0.1, 0.2, 0.3. Based on the maximum observed correlations within the data larger values for ρ would be redundant. For the weighing function inside the fusion penalty we selected $w(r) = |r|$. In the adaptive SVS methods we used the univariate OLS-based weights obtained with f equal to the mean and applied the power transformations to them via

$$\pi_k^{\nu} = d_k \left(\sum_l b_{kl}^{\text{ols}}/p \right)^{-\nu}, \tag{6}$$

where $\nu = 1, 2, 0.5$ with d_k chosen so that $\sum_k \pi_k = q$. We respectively denote the resulting methods corresponding to each value of ν as L_2- aSVS(1), L_2- aSVS(2), and L_2- aSVS(0.5) for $\alpha = 2$, and $L_\infty-$ aSVS(1), $L_\infty-$ aSVS(2), and L_∞ $-$ aSVS(0.5) for $\alpha = \infty$.

In the comparison of methods we also include the lasso and the adaptive lasso methods. We used four different variants of the adaptive lasso described in (A.1), which differed by the choices of weights w_{kl}. The first three variants utilized the same univariate OLS approach as the adaptive SVS methods, which was achieved by putting $w_{kl} = \pi_k^{\nu}$ for all $l = 1, \ldots, p$ with the same choices for $\nu = 1, 2, 0.5$. The corresponding adaptive lasso methods are referred to as alasso(1), alasso(2), and alasso(0.5). The fourth variant of the adaptive lasso utilized weights based on the lasso with λ determined through CV. We refer to the resulting method as alasso(lasso).

3.3. Measures of performance

In order to judge the performances of each method we calculated and plotted several measures, which focus on the quality of an estimator with respect to three different perspectives. Our first measure of quality of an estimate is the fraction of non-zero values among estimates of true zero coefficients, also known as the *false selection rate* (*FSR*). A complementary measure is the ratio of non-zero values among estimates of the true non-zero coefficients, also known as the *true selection rate* (*TSR*).

The overall quality of an estimate $\widehat{\boldsymbol{B}} = (b_{ij})$ can be measured by the *squared expectation prediction error* $SEPE = \|\boldsymbol{X}\widehat{\boldsymbol{B}} - \boldsymbol{X}\boldsymbol{B}\|^2$. Since such definition of prediction error is sample size dependent, for comparison of prediction errors over various sample sizes it is useful to scale *SEPE* by n, which yields the average squared prediction error per individual. We use such rescaled *SEPE* to compare the performances under the scenarios B and C. For the comparison of the methods we first judge the methods based on how low their minimum observed values of *SEPE* are.

Additionally, we use the *total estimation error* (*TEE*), which is the ℓ_1 distance between the estimates and the true values of the regression matrix $\boldsymbol{B}$. Focusing on all SNPs in Q, the causal SNPs in Q_1, or the non-causal SNPs in Q_0, respectively, leads to

$$TEE_z^{\beta} = \sum_{k \in Q_0} \sum_{j=1}^{p} |\, b_{kj}|, \quad TEE_{nz}^{\beta} = \sum_{k \in Q_1} \sum_{j=1}^{p} |\, b_{kj} - \beta_{kj}|,$$

$$TEE^{\beta}_{\text{sum}} = TEE^{\beta}_{z} + TEE^{\beta}_{nz}.$$

3.4. Investigation under scenario A

We first focus on the performance aspects of the methods under scenario A. The choice of penalty parameters were determined based on 2-fold CV. Analysis showed that this is a reliable procedure under scenarios similar to A (results not shown). Table 2 illustrates the performance of the estimation methods.

3.4.1. Performance of L_2- aSVS and $L_\infty-$ aSVS methods

In Table 2 we see that, when compared to the non-adapted L_2- SVS, all variants of L_2- aSVS provide noticeably better performances in terms of minimum *SEPE* and *FSR* and *TSR*. The improvement over the non-adaptive SVS is most visible for L_2- aSVS(2), while L_2- aSVS(1) is a very close second. Besides good prediction performance, the adaptive SVS variants yield similarly flattering results in terms of *FSR*. In addition, the comparison of the true selection rates of L_2- aSVS(2), L_2- aSVS(1), L_2- aSVS(0.5) against L_2- SVS shows appreciable improvements by the adaptive SVS methods over the (non-adaptive) SVS.

Turning to methods $L_\infty-$ SVS and $L_\infty-$ aSVS, we observe a similar rate of improvement in terms of *SEPE*, *FSR* and *TSR* by the adaptive SVS methods over the non-adaptive $L_\infty-$ SVS (relative to the ℓ_2 methods). Similarly to the ℓ_2 methods, the ℓ_∞ adaptive SVS methods provide substantial improvement of prediction errors over their non-adaptive counterpart. Generally speaking, the ℓ_∞ methods show noticeably higher *FSR*. In terms of *TSR* the ℓ_∞ methods exhibit as good a performance as their ℓ_2 counterparts, when they successfully identify all non-zero parameters in all data sets. The non-adaptive SVS methods appear to be inferior to the adaptive SVS in this respect.

Next, we focus on the performances of the GFlasso and the lasso relative to the adaptive SVS methods. It appears that the adaptive SVS methods quite convincingly outperform the rest of the considered methods. Compared to any of the adaptive SVS methods, both the lasso and the variants of the GFlasso yield substantially inferior performance in terms of the three measures. This suggests that the exploitation of the correlation in the response by

Table 2. Average observed *SEPE*, *FSR*, *TSR* at cross-validated penalties under scenario *A*.

Method	λ_{cv} (μ_{cv})	*SEPE*	*FSR*	*TSR*
L_2- aSVS(2)	190	***2.73***	***0.0038***	***1.00***
$L_\infty-$ aSVS(2)	320	3.03	0.028	***1.00***
L_2- aSVS(1)	34	3.26	0.011	***1.00***
$L_\infty-$ aSVS(1)	55	4.12	0.039	***1.00***
L_2- aSVS(0.5)	20	4.47	0.021	***1.00***
$L_\infty-$ aSVS(0.5)	35	5.50	0.046	***1.00***
L_2- SVS	17	7.58	0.031	0.97
$L_\infty-$ SVS	33	8.97	0.047	0.97
GFlasso($\rho = 0.2$)	8 (65)	9.70	0.073	0.80
GFlasso($\rho = 0.3$)	11 (40)	10.43	0.024	0.71
GFlasso($\rho = 0.1$)	6 (50)	10.24	0.150	0.90
GFlasso($\rho = 0.05$)	5 (70)	10.45	0.240	0.99
lasso	11	11.47	0.015	0.67

Note: The averages are taken over the 25 data sets and the rows are ordered by the minimum *SEPE* with the best values in each column marked by bold italic.

the fusion penalty of the GFlasso is somewhat limited and generally performs worse than what is achieved by the ℓ_2 and ℓ_∞ penalties used by the adaptive SVS methods. Overall, it seems that L_2- aSVS(1) and L_2- aSVS(2) are the champions among all of the considered methods under scenario A.

3.5. Performance for a single data set under scenario A

For each data set under scenario A we obtained estimates for $\boldsymbol{B} = (\beta_{ij})$ using 2-fold cross-validated values of the penalty parameters. In Table 3 we included the observed values of *SEPE*, *FSR*, *TSR*, TEE_z^β, TEE_{nz}^β and TEE_{sum}^β. In this comparison we included the naive OLS in order to compare this simplistic method with the more sophisticated methods.

When comparing the adaptive SVS under cross-validated penalties among themselves, it seems to be L_2- aSVS(2) and $L_\infty-$ aSVS(2) that are ahead of the other methods in terms of *FSR*. Both of these methods falsely select only 21 out of the 3000 non-causal SNPs. Additionally, the other variants of the adaptive SVS are not trailing too much behind. The results are quite impressive, especially when compared with the GFlasso and the lasso methods. The impressiveness of the low *FSR* by L_2- aSVS(2) and $L_\infty-$ aSVS(2) is further enhanced by the fact that virtually all of the falsely selected SNPs have estimates that are smaller than all of the estimates for the causal SNPs. Moreover, all of the causal SNPs were correctly selected by the adaptive SVS methods. Compared with the adaptive SVS, the only variant of the GFlasso that bears any comparison is that with $\rho = 0.3$. In terms of prediction error, it is again L_2- aSVS(2) and $L_\infty-$ aSVS(2) that perform best among all considered methods. The other adaptive methods also provide considerably improved performance over the two non-adaptive SVS methods and especially over the GFlasso and the lasso. The naive OLS method performs relatively poorly at both estimation and model selection.

3.6. Investigation under scenarios B and C

The data sets simulated under scenarios B and C were analysed by the methods employed under scenario A and by the additional four variants of the adaptive lasso. The aspect of performance we investigated under scenarios B and C was the sensitivity of each method's

Table 3. Observed values of performance measures *SEPE*, *FSR*, *TSR*, TEE_z^β, TEE_{nz}^β and TEE_{sum}^β for a single data set under scenario *A*.

Method	Penalty	*SEPE*	*FSR*	*TSR*	TEE_z^β	TEE_{nz}^β	TEE_{sum}^β
L_2- aSVS(2)	$\lambda_{cv} = 190$	***2.73***	***0.006***	***1***	0.18	3.6	3.78
$L_\infty-$ aSVS(2)	$\lambda_{cv} = 280$	3.03	0.041	***1***	0.24	3.8	4.04
L_2- aSVS(1)	$\lambda_{cv} = 34$	3.26	0.012	***1***	0.20	3.5	3.7
$L_\infty-$ aSVS(1)	$\lambda_{cv} = 55$	4.12	0.043	***1***	0.33	3.7	4.03
L_2- aSVS(0.5)	$\lambda_{cv} = 20$	4.47	0.019	***1***	0.25	3.2	3.45
$L_\infty-$ aSVS(0.5)	$\lambda_{cv} = 35$	5.50	0.053	***1***	0.32	3.4	3.72
L_2- SVS	$\lambda_{cv} = 17$	7.58	0.029	***1***	0.28	2.4	2.68
$L_\infty-$ SVS	$\lambda_{cv} = 31$	8.92	0.029	***1***	***0.16***	2.6	2.76
GFlasso($\rho = 0.2$)	$\lambda_{cv} = 8.4$, $\mu_{cv} = 64.9$	9.70	0.250	***1***	0.44	2.3	2.74
GFlasso($\rho = 0.05$)	$\lambda_{cv} = 4.6$, $\mu_{cv} = 68.7$	10.45	0.690	***1***	0.87	2.8	3.67
GFlasso($\rho = 0.1$)	$\lambda_{cv} = 6.4$, $\mu_{cv} = 52.3$	10.24	0.420	***1***	0.66	2.6	3.26
GFlasso($\rho = 0.3$)	$\lambda_{cv} = 10.7$, $\mu_{cv} = 38.3$	10.43	0.094	***1***	0.37	***1.9***	***2.27***
lasso	$\lambda_{cv} = 11$	11.47	0.059	***1***	0.44	***1.9***	2.34
naive OLS (MTC)	none	15.85	0.016	0.75	4.20	4.5	8.70

Note: The rows are ordered according to observed *SEPE* and the best value in each column is marked by bold italic font.

performance to sample size. We used the $N = 25$ simulated genotype-phenotype data sets under each scenario, which we repeatedly analyzed using $k \leq 2000$ of the 2000 individuals, where k ranged between 400 and 2000 in steps of 100. When increasing k, a randomly selected additional 100 individuals were added to the previously selected individuals in each step.

We additionally used the adaptive lasso methods alasso(1), alasso(2), alasso(0.5) and alasso(lasso). In the GFlasso methods we put ρ equal to 0.05, 0.1, 0.2, 0.3 and 0.4 under scenario B, whereas under scenario C we only considered values 0.05, 0.1 and 0.2. These choices were motivated by the maximum absolute values of correlation between the components of the phenotypes in each of these data sets, which under scenario B were between 0.5 and 0.6, while under scenario C they were between 0.2 and 0.3 for all considered sample sizes. The observed correlation coefficients are plotted in Figure 3, which shows histograms of the correlations and the maximum absolute value of the correlations as function of sample size. The plots clearly show the effect of the parameter choices we made under the two scenarios. In the plots on the right we can see the maxima of absolute correlations, which decrease with sample size. We can also clearly see where the maxima lie, which justifies the chosen values for ρ under the two scenarios.

For each sample size, scenario and method we calculated an estimate of $\boldsymbol{B}$ and used these to obtain average values of the prediction error *SEPE*, selection rates *FSR*, *TSR*, and total estimation errors TEE_z^{β}, TEE_{nz}^{β}, TEE_{sum}^{β}, which we plotted in Figure 4. Since there are large differences between the performance of different methods, the plots are on logarithmic scales. We also note that we plotted smoothed versions of the actually observed lines.

3.6.1. Selection rates

Judging from the plots in Figure 4, while keeping in mind the natural trade-off between *FSR* and *TSR*, it is quite clear that the variants of the *adaptive* SVS perform very well in terms of both *FSR* and *TSR* under both scenarios. Especially L_2- aSVS(1), L_2- aSVS(2), L_2- aSVS(0.5) and $L_\infty-$ aSVS(1) exhibit *FSR* below 1% and *TSR* generally well above 50% for (almost) all sample sizes. Additionally, with *TSR* increases up to about 80% with sample size for these methods, which under scenario C makes them "catch up" to the best performing method in terms of *TSR*, while under scenario B they do not trail far behind the champions either. The most positive aspect of the good *TSR* performance by the adaptive SVS methods is the fact that it is not paid for by a lousy *FSR* showing. For all adaptive SVS methods *FSR* remains well controlled for all considered sample sizes.

As far as the non-adaptive SVS methods are concerned, both of them appear to be very good performers in terms of *FSR*, lacking only slightly behind the adaptive SVS methods under both scenarios. In terms of *TSR* both the non-adaptive SVS methods are performing well under scenario B, while under scenario C the method L_2- SVS appears to be noticeably inferior compared to the other SVS methods.

We notice large differences among the variants of GFlasso, where the performances in terms of *FSR* and *TSR* hugely depend on the value of ρ. Looking at the *FSR* plots, it is clear that for small values of ρ the performance of GFlasso in terms of *FSR* is quite bad under both scenarios, where especially under scenario C two out of the three of the GFlasso variants have an unacceptably high *FSR*.

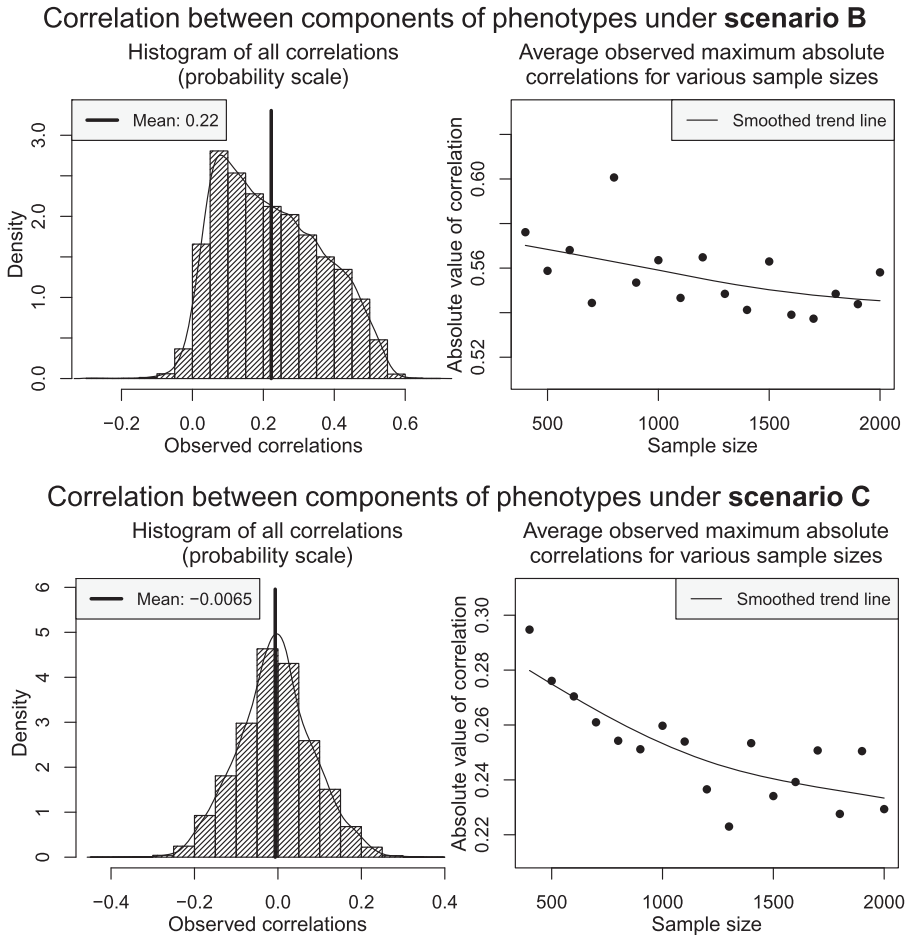

Figure 3. Sample correlation coefficients between components of the phenotype under scenarios B and C. The histograms show the observed correlations between the 10 components of the phenotypes for 25 data sets and all considered sample sizes (a total of 340 values) with corresponding kernel density estimators and mean correlations superposed onto the histogram as solid black lines. The curves (right) show the mean maximum absolute value of correlations as functions of the sample sizes (means taken over the 25 data sets).

Turning to the adaptive lasso, Figure 4 shows that in terms of *FSR* the differences between the four variants of the adaptive lasso are rather small. Keeping in mind the log-scales of the *y*-axes, there seems to be virtually no difference between the methods. It is quite clear that the adaptive lasso does not substantially improve on the lasso.

3.6.2. Prediction and estimation errors

Looking at the *SEPE* plot (top) in Figure 4 we can clearly see that it is again the adaptive methods that provide the superior performances in terms of *SEPE*, where L_2-aSVS(2), L_2-aSVS(1), $L_\infty-$aSVS(2) and $L_\infty-$aSVS(1) particularly stand out under both scenarios. Under both scenarios the ratio of improvement was increasing with sample size suggesting that the adaptive SVS methods make more efficient use of the additional data. In

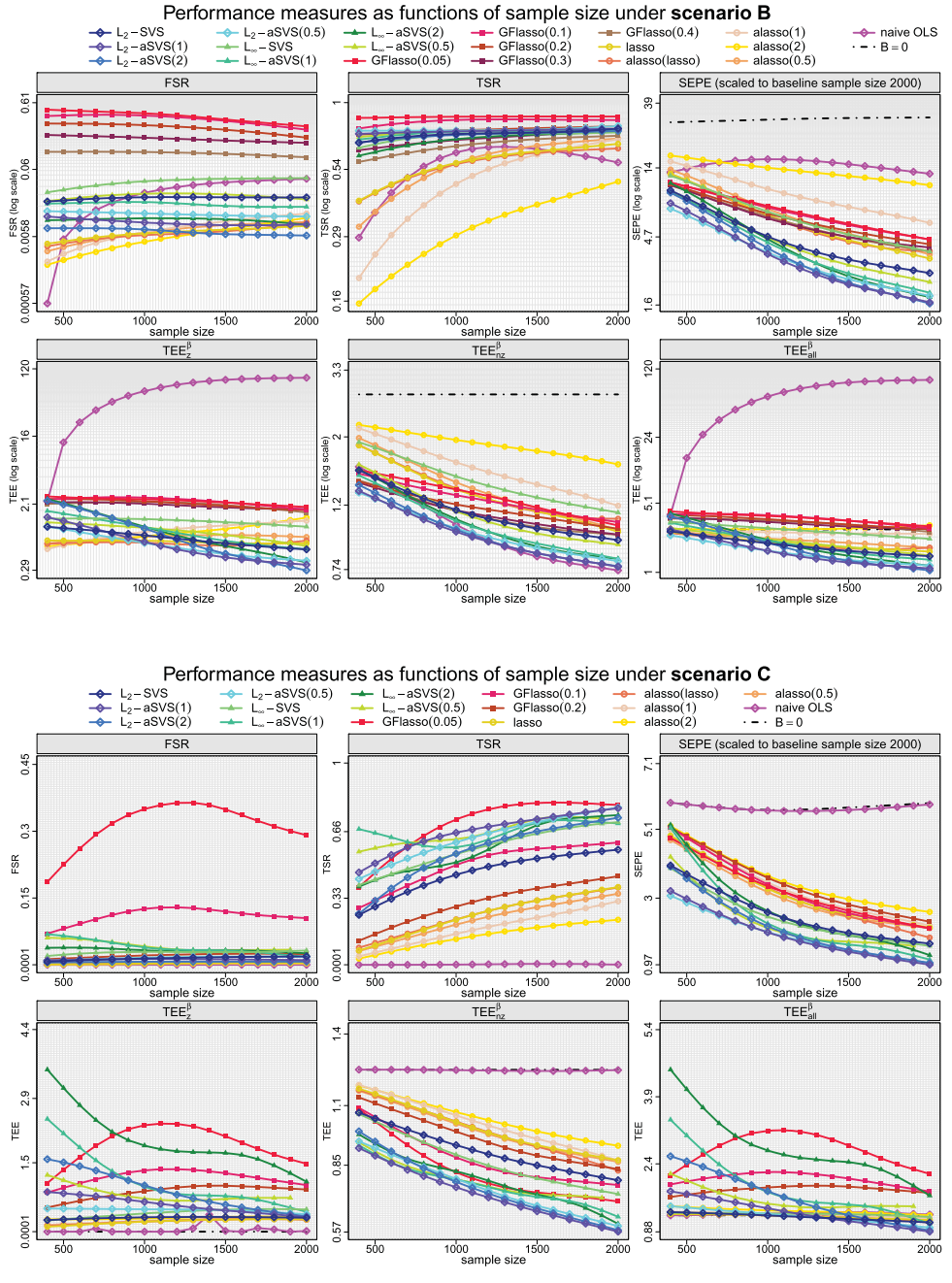

Figure 4. Observed average measures of performance *FSR*, *TSR*, *SEPE* and TEE_z^{β}, TEE_{nz}^{β}, TEE_{sum}^{β} for various methods (including the all-zero estimate $\widehat{\boldsymbol{B}}_0 = 0$) as functions of sample sizes under scenarios B (top) and C (bottom). For easier reading the values of *SEPE* were scaled up by the base line sample size of 2000. The *y*-axes are on natural logarithmic scales.

Figure 4 we present three total estimation error measures for each scenario, namely TEE_z^{β}, TEE_{nz}^{β}, TEE_{sum}^{β}. Looking at the overall estimation error TEE_{sum}^{β} under scenario B we notice an ordering of the methods that is very similar to that given by *SEPE*. It is again clear that

the best performance is provided by the adaptive SVS methods. The plots with TEE_z^β and TEE_{nz}^β provide further confirmation of the superiority of the adaptive SVS methods. Turning to scenario C, the overall estimation error TEE_{sum}^β shows that the best choices seem to again be the three adaptive SVS methods L_2- aSVS(1), L_2- aSVS(2) and L_2- aSVS(0.5), while $L_\infty-$ aSVS(1), L_2- aSVS(2) and the non-adaptive L_2- SVS are not far behind.

4. Application to data: alternative splicing

In this section we present the results of an eQTL analysis of the expression data generated by the *Geuvadis RNA sequencing project for 1000 Genomes samples* [14]. The goal is to identify a SNP-driven gene expression regulation process known as *alternative splicing*. About 94% of our genes are so called *interrupted genes* [26], which means that they consist of several regions of different functional type referred to as *exons* and *introns*. The number of exons in human genes varies between 1 and 363 and the average number of exons per gene is about 10 [21]. During DNA transcription the genetic code undergoes a process called *splicing*, when introns are removed while exons are preserved and transcribed into RNA (i.e. *expressed*). Crucially, however, not all exons are always expressed, which means that the same genetic code in a gene can lead to different RNA transcripts. This occurs when, during RNA transcription, different subsets of exons are expressed. This phenomenon when a single gene produces different RNA transcripts is called *alternative splicing*. Interestingly, different RNA transcripts do not have to result in differential protein expression.

If expression data per exon is available, it is possible to look for evidence of alternative splicing. This can be done by checking if all exons are observed in equal proportion. There are technical limitations that need to be taken into account during such analysis. For instance, exon length may affect measurements since exons below a certain minimal length may be less efficiently handled both during sequencing and during alignment. This effect is not linear, which means that as exon length decreases, the number of reads mapping to the exon decreases with exon length faster than linearly.

4.1. Data

The complete Geuvadis data contains 148,002 exons spread over 15,480 genes. After quality control (QC) performed by the Geuvadis team there were 462 unrelated individuals from various cohorts remaining. In the pilot study we restricted the focus only to exons on chromosome 1 and used the usual QC criteria such as a MAF threshold of 5%, etc. The distribution of per-gene exon counts on chromosome 1 in the raw data is shown in Figure 5. While the maximum observed number of exons in a single gene was 105, over 50% of the included genes have 8 or fewer exons, only 10% have more than 20 exons and only 1% genes contained more than 40 exons.

The raw data was pre-processed and to correct for large trends typically produced by batch effects, the first 10 principal components were removed. However, in the pilot we did not correct for the effect of exon length on expression. Exons that were not expressed at all in the data were eliminated. Moreover, genes with only a single exon were also removed, since these cannot undergo alternative splicing. From the total of 14,758 exons

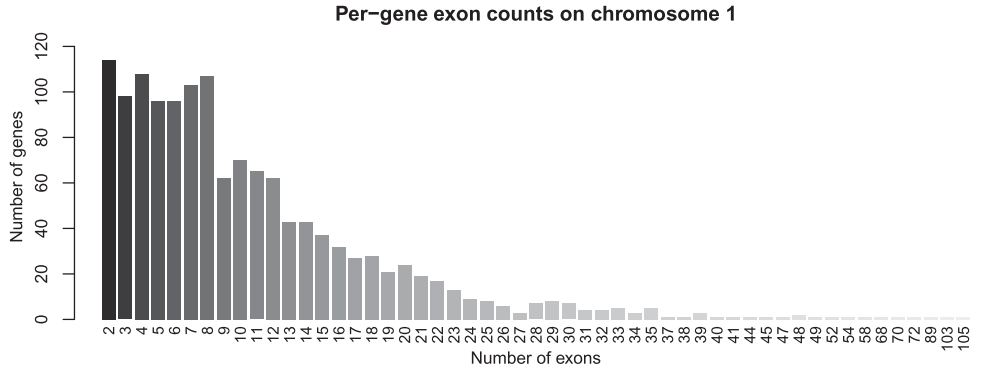

Figure 5. Distribution of per-gene exon counts (top) and per-gene SNP counts (bottom) on chromosome 1 in pre-processed Geuvadis data. Shades of gray correspond to the height of each bar in both plots.

in 1389 genes on chromosome 1, there were 14,656 exons in 1376 genes left in the data set after pre-processing. In order to improve the homogeneity of the data we additionally removed the data for the Yoruba population from the pilot study, after which there were 373 samples left. Besides removing the top 10 principal components, the exon expression data was further transformed to account for the fact that exon expressions are on an exponential scale. The transformation used was the variance-stabilizing transformation $h(y) = \gamma \arcsin h(a + by)$ [11]. The constants a, b, γ are determined by fitting an assumed quadratic relationship between the mean and the variance of the data parameterized via a, b, γ (for details see [11]).

4.2. Data analysis

The transformed expression counts for each exon of the gene were entered as multivariate response matrix $\boldsymbol{Y}$ into a linear model of (1) with individual SNP genotypes forming the columns of the design matrix $\boldsymbol{X}$. For every gene the number of rows in both $\boldsymbol{Y}$ and $\boldsymbol{X}$ (i.e. the sample size) was the number of individuals in the data, which was $n = 373$. The number of columns in $\boldsymbol{Y}$ and $\boldsymbol{X}$ differed between the genes. For simplicity of analysis each column of $\boldsymbol{Y}$ and $\boldsymbol{X}$ were centered with the corresponding (per-column) sample means.

For each of the penalized methods the values of penalty parameter was optimized via 10-fold CV. The estimation methods of primary interest were the variants of the L_2-aSVS method. For reasons of computational complexity we chose the ℓ_2-based methods instead of the ℓ_∞ methods, since the available solver MOSEK is considerably less efficient than the `glmnet` package for R. For the analysis we used both the non-weighted (L_2-SVS) and weighted variants of the method with several different weighting schemes, which were all based on the univariate OLS approach. Three of them were identical to the ones used throughout Chapter 2.3, which we denoted by L_2-aSVS(1), L_2-aSVS(2) and L_2-aSVS(0.5). In addition to those we also considered a number of higher power transformations of the OLS-based weights and denote the corresponding methods as L_2-aSVS(ν), where $\nu = 3, \ldots, 10$ (see (6)). The higher power transformations put more emphasis on

Table 4. Comparison of SNP and gene selection counts by various regression methods.

	genes (total of 1376)		SNPs (total of 4,958,575)		
method (M)	selected	ratio	selected	ratio	avg per-gene
L_2-SVS	512	37.21%	1782	0.0359%	3.48
L_2-aSVS(0.5)	442	32.12%	1399	0.0282%	3.17
L_2-aSVS(1)	368	26.74%	1100	0.0222%	2.99
L_2-aSVS(2)	280	20.35%	797	0.0161%	2.85
L_2-aSVS(3)	235	17.08%	662	0.0134%	2.82
L_2-aSVS(4)	205	14.90%	644	0.0130%	3.14
L_2-aSVS(5)	200	14.53%	653	0.0132%	3.27
L_2-aSVS(6)	205	14.90%	689	0.0139%	3.36
L_2-aSVS(7)	211	15.33%	741	0.0149%	3.51
L_2-aSVS(8)	216	15.70%	815	0.0164%	3.77
L_2-aSVS(9)	231	16.79%	996	0.0201%	4.31
L_2-aSVS(10)	222	16.13%	1033	0.0208%	4.65
lasso	80	5.81%	231	0.0047%	2.89
alasso(0.5)	82	5.96%	253	0.0051%	3.09
alasso(1)	86	6.25%	275	0.0055%	3.20
alasso(2)	108	7.85%	345	0.0070%	3.19
naiveols	242	17.59%	6669	0.1345%	27.56

the initial univariate OLS estimates. It seems reasonable to expect that there is an optimum transformation which strikes the right balance between the information contained within the univariate OLS estimates, which determine the weights, and the ability of penalized regression methods to uncover the association. Analogously to the adaptive SVS methods above, we denote the three adaptive lasso variants as alasso(1), alasso(2), alasso(0.5).

For the SVS methods Table 4 shows that the non-adaptive L_2-SVS yields the least sparse solution and selects 1782 SNPs in 512 genes. With the adaptation we obtain much sparser solutions and both the SNP and the gene selection counts decrease. Since the 11 adaptive SVS methods differ only by the degree to which the initial univariate OLS estimates shape the final solution it is not surprising that higher values of p lead to more sparse solutions with for instance L_2-aSVS(1) and L_2-aSVS(2). The sparsity generally increases further with increasing p.

Finally, we also focused on the comparison of the individual penalized regression methods against each other in applied setting. It turned out, likely due to the more efficient exploitation of the multivariate nature of the responses, that the adaptive SVS methods provided SNP selection for a significantly larger portion of the considered genes compared to the lasso and the adaptive lasso variants. It seems that the latter are perhaps too restrictive during selection. While selecting a larger number of genes, the adaptive SVS variants simultaneously limit the number of selected SNPs to a manageably small number.

5. Discussion

In this paper we presented an adaptive SVS, which is a method for estimating parameters in the multivariate multiple regression model of (1) particularly suitable for applications where the assumptions of *sparsity* and *common association* are reasonable. We put the method under thorough scrutiny in a realistic simulation study in the context of

genotype-phenotype data, where we compared it from numerous perspectives with several other methods under several different simulation scenarios. We considered several flavors of the adaptive SVS method which differed by the type of penalty and by the way the adaptation weights were calculated (i.e. the specific choices of α and π_k in Equation (3)). A general conclusion that can be drawn from our investigation is that the adaptive SVS method is a powerful tool and many of the considered flavors yield good performance in terms of both selection rates, prediction errors and estimation accuracy. An overall impression is that most of the considered flavors of the adaptive SVS method perform persuasively better than the other considered methods. This includes even those methods that were specifically tailored for the multivariate-response model with correlated components.

Due to the popularity of the naive OLS, we also considered the performance of this method under several different scenarios. We showed that although the method can work quite well in certain respects such as TEE^{β}_{nz} under scenario B, it often fails miserably in other respects (TEE^{β}_{z} and *SEPE*) and/or under other scenarios (A and C). However, the adaptive SVS methods seem to be on par with the naive OLS even under the scenario favorable to the naive OLS, while the naive OLS is clearly inferior to the adaptive SVS methods in all of the other considered measures. It seems therefore clearly unwise to use the naive OLS as the method of choice for such analysis.

The adaptive SVS method requires user input in two distinct ways. On the one hand, it requires the user to select a suitable way to determine the adaptation weights, where we showed that an approach as simple as univariate regression can already yield very favorable behavior. This suggests that there is still room for improvement of the performance of the adaptive SVS method by using a more sophisticated way of determining adaptation weights although probably at the cost of increased computational burden. On the other hand, like any penalized regression method, the adaptive SVS also requires a good choice of value of the tuning penalty parameter.

An advantage of using the univariate OLS for determining the adaptation weights π_k is the straightforwardness of such approach. It relies on linear correlation between regressors and responses, which also makes it simple and fast to implement. After a fixed f is chosen, there are no more tuning parameters involved in determining the adaptation weights. This increases the appeal of the univariate OLS approach in comparison with for instance using the non-adaptive SVS or the lasso as the basis for determining the weights, which would require additional analysis to determine their penalty parameters.

GFlasso represents a natural extension of the lasso, which seeks to improve on the lasso in the current model by utilizing the covariance structure of the multivariate response by introducing a second penalty, although this comes at a price of higher computational complexity as well as a more difficult choice of tuning penalty parameters λ and μ. The SVS methods, on the other hand, require the choice of only one penalty parameter, while also reflecting the multivariate nature of the response to some extent.

Usual ways to determine the tuning parameters include CV and information criteria such as AIC, BIC and GIC [7]. In our analysis we showed that the non-adaptive approach based on CV works quite well towards allowing the method to maximize its potential.

In summary, the adaptive SVS is a strong method that in our opinion should become the workhorse for analysis of association between a large number of regressors and correlated multivariate phenotypes.

Acknowledgments

We wish to thank Dr. Renee de Menezes of VU University Medical Center, the Netherlands, for pre-processing the Geuvadis data and for providing the corresponding R code.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

J. Pecanka http://orcid.org/0000-0002-6857-6001
A. W. van der Vaart http://orcid.org/0000-0002-8074-2375
M. A. Jonker http://orcid.org/0000-0003-0134-8482

References

[1] D.B. Allison, B. Thiel, P. St. Jean, R.C. Elston, M.C. Infante, and N.J. Schork, *Multiple phenotype modeling in gene-mapping studies of quantitative traits: Power advantages*, Am. J. Hum. Genet. 63 (1998), pp. 1190–1201.
[2] P. Bickel, Y. Ritov, and A. Tsybakov, *Simultaneous analysis of Lasso and Dantzig selector*, Ann. Stat. 36 (2008), pp. 1567–1594.
[3] P. Buhlmann and S. van de Geer, *Statistics for High-Dimensional Data. Methods, Theory and Applications*, 1st ed., Springer-Verlag, Berlin, Heidelberg, 2011.
[4] P. Bühlmann, P. Rütimann, S. van de Geer, and C. Zhang, *Correlated variables in regression: Clustering and sparse estimation*, J. Statist. Plann. Inference. 143 (2013), pp. 1835–1858.
[5] I. Castillo, J. Schmidt-Hieber, and A. van der Vaart, *Bayesian linear regresssion with sparse priors*, Ann. Stat. 43 (2015), pp. 1986–2018.
[6] I. Castillo and A. van der Vaart, *Needles and straw in a haystack: Posterior concentration for possibly sparse sequences*, Ann. Stat. 40 (2012), pp. 2069–2101.
[7] Y. Fan and C.Y. Tang, *Tuning parameter selection in high dimensional penalized likelihood*, JRSS Ser. B 75 (2013), pp. 531–552.
[8] T.E. Galesloot, K. van Steen, L.A.L.M. Kiemeney, L.L. Janss, S.H. Vermeulen, and Y.S. Aulchenko, *A comparison of multivariate genome-wide association methods*, PLoS. ONE. 9 (2014), p. e95923. doi:10.1371/journal.pone.0095923.
[9] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed., Springer Series in Statistics, Springer, New York, 2009. ISBN 978-0-387-84857-0. doi:10.1007/978-0-387-84858-7. Data mining, inference, and prediction.
[10] J. Huang, S. Ma, and C. Zhang, *Adaptive lasso for sparse high-dimensional regression models*, Statist. Sinica 18 (2008), pp. 1603–1618.
[11] W. Huber, A. von Heydebreck, H. Sultmann, A. Poustka, and M. Vingron, *Variance stabilization applied to microarray data calibration and to the quantification of differential expression*, Bioinformatics 18 (2002), pp. S96–S104. 10.1093/bioinformatics/18.suppl_1.S96.
[12] S. Kim, K. Sohn, and E.P. Xing, *A multivariate regression approach to association analysis of quantitative trait network*, Bioinformatics 25 (2009), pp. 204–212.
[13] S. Kim, E.P. Xing, and J.D. Storey, *Statistical estimation of correlated genome association to a quantitative trait network*, PLoS. Genet. 5 (2009), p. e1000587.
[14] T. Lappalainen, M. Sammeth, M.R. Friedländer, P.A.C. 't Hoen, J. Monlong, M.A. Rivas, M. Gonzàlez-Porta, N. Kurbatova, T. Griebel, P.G. Ferreira, M. Barann, T. Wieland, L. Greger, M. van Iterson, J. Almlöf, P. Ribeca, I. Pulyakhina, D. Esser, T. Giger, A. Tikhonov, M. Sultan, G. Bertier, D.G. MacArthur, M. Lek, E. Lizano, H.P.J. Buermans, I. Padioleau, T. Schwarzmayr, O. Karlberg, H. Ongen, H. Kilpinen, S. Beltran, M. Gut, K. Kahlem, V. Amstislavskiy, O. Stegle, M. Pirinen, S. B. Montgomery, P. Donnelly, M. I. McCarthy, P. Flicek, T.M. Strom, H. Lehrach, S. Schreiber, R. Sudbrak, Á Carracedo, S.E. Antonarakis, R. Häsler, A.-C. Syvänen,

G.-J. van Ommen, A. Brazma, T. Meitinger, P. Rosenstiel, R. Guigó, I.G. Gut, X. Estivill, and E. T. Dermitzakis, *Transcriptome and genome sequencing uncovers functional variation in humans*, Nature 501 (2013), pp. 506–511.
[15] D. Malioutov, M. Cetin and A.S. Willsky, *Sparse signal reconstruction perspective for source localization with sensor arrays*, IEEE. Trans. Signal Process. 53 (2005), pp. 3010–3022.
[16] S. Negahban and M.J. Wainwright, *Simultaneous support recovery in high dimensions: Benefits and perils of block ℓ_1/ℓ_∞-regularization*, Technical report, Department of Statistics, UC Berkeley, CA, US, 2011.
[17] G. Obozinski, M.J. Wainwright, and M.I. Jordan, *Support union recovery in high-dimensional multivariate regression*, Ann. Stat. 39 (2011), pp. 1–47.
[18] J. Pecanka, *Multi-step statistical methods for simultaneous inference in genetics*, Ph.D. thesis, Vrije Universiteit, Amsterdam, The Netherlands, 2016
[19] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2015. Available at https://www.R-project.org/.
[20] T Similä and J. Tikka, *Input selection and shrinkage in multiresponse linear regression*, Comput. Statist. Data. Anal. 52 (2007), pp. 406–422.
[21] T. Strachan and A. Read, *Human Molecular Genetics*, 4th ed., Taylor & Francis Group, New York, 2011.
[22] R. Tibshirani, *Regression shrinkage and selection via the lasso*, JRSS Ser. B 58 (1996), pp. 267–288.
[23] J.A. Tropp, A.C. Gilbert, and M.J. Strauss, *Algorithms for simultaneous sparse approximation*, Signal Process. 86 (2006), pp. 572–602. Special issue on 'Sparse approximations in signal and image processing'.
[24] B.A. Turlach, W.N. Venables, and S.J. Wright, *Simultaneous variable selection*, Technometrics 47 (2005), pp. 349–363.
[25] H. Wang and C. Leng, *A note on adaptive group lasso*, Comput. Statist. Data Anal. 52 (2008), pp. 5277–5286.
[26] A.J. Ward and T.A. Cooper, *The pathobiology of splicing*, J. Pathol. 220 (2010), pp. 152–163.
[27] F. Wei and J. Huang, *Consistent group selection in high-dimensional linear regression*, Bernoulli 16 (2010), pp. 1369–1384.
[28] F. Wei, J. Huang, and H. Li, *Variable selection and estimation in high-dimensional varying-coefficient models*, Statist. Sinica 21 (2011), pp. 1515–1540.
[29] M. Yuan and Y. Lin, *Model selection and estimation in regression with grouped variables*, JRSS Ser. B 68 (2006), pp. 49–67.
[30] C.-H. Zhang and J. Huang, *The sparsity and bias of the lasso selection in high-dimensional linear regression*, Ann. Stat. 36 (2008), pp. 1567–1594.
[31] H. Zou, *The adaptive lasso and its oracle properties*, JASA 101 (2006), pp. 1418–1429.
[32] H. Zou and T. Hastie, *Regularization and variable selection via the elastic net*, JRSS Ser. B 67 (2005), pp. 301–320.

Appendix

A.1 Software

The GFlasso estimates were computed using a MATLAB (www.matlab.com) script, which was graciously provided to us by Seyoung Kim, one of the authors of the method. Estimates of regression parameters by the ℓ_2 norm-based adaptive and non-adaptive SVS methods were obtained using the R package `glmnet` [19], while the ℓ_∞ norm-based estimates were determined using the large-scale optimization software MOSEK and the R interface package `Rmosek` available from CRAN. MOSEK is a commercial high performance software for large-scale optimization. A free academic licence can be obtained at http://www.mosek.com. Finally, the lasso and the adaptive lasso estimates we calculated using `glmnet`. For the aSVS method a custom script was used to facilitate the usage of the existing software (e.g. glmnet, MOSEK). This script has not been implemented as a stand-alone package.

In terms of computational burden, the adaptation in aSVS requires essentially only the fitting of a large number of *univariate* regressions, which, compared to the non-adaptive SVS, does not add much computational complexity in typical applications. However, the computational efficiency of aSVS depends highly on the specific algorithm and solver used. The availability of such algorithm and the efficiency of the utilized solver closely links to the choice of α in Equation (3). For $\alpha = 1$ and $\alpha = 2$ the method essentially reduce to the lasso and the group lasso, respectively, for both of which there exist very efficient algorithms and solvers (e.g. glmnet, [19]). Consequently, on an ordinary machine, an application of aSVS to a single dataset of the type considered in this paper is a matter of seconds, with the exact time depending on the hardware. For our third choice $\alpha = \infty$, the situation is somewhat less favorable, as no specialized solver appears to be available at this time. Our implementation using the general solver MOSEK requires needed minutes rather than seconds, making this choice less suitable for application in the very high-dimensional settings.

A.2 Adaptive lasso

As an extension to the lasso, the *adaptive lasso* [31] is defined as

$$\widehat{\boldsymbol{B}}_{\text{alasso}}(\lambda) = \underset{\boldsymbol{B}=(\beta_{ij})}{\arg\min} \|\boldsymbol{Y} - \boldsymbol{XB}\|_2^2 + \lambda \sum_{i,j} w_{ij}|\beta_{ij}|. \tag{A1}$$

The difference between the penalty of the lasso and that of the adaptive lasso is the presence of weights w_{ij} in the latter. As we can see from (A1), employing the weights inside the ℓ_1-type penalty allows the user to differentiate the amount penalization each regressor receives, thus permit some of them to obtain larger estimates (in absolute value sense) compared to the lasso while forcing the rest closer to zero.

A.3 Graph-guided fused lasso

The *graph-guided fused lasso* (GFlasso) method is based on the lasso with a secondary *fusion penalty* introduced to bind the estimates of parameters of the same regressor when highly correlated responses are modeled [12,13]. Their *graph-guided fusion penalty* is guided by a *phenotype graph*. More specifically, using correlations of the phenotypes a graph of phenotypes as nodes is constructed in which two phenotypes are connected by an edge if their correlation exceeds a preset bound. Whenever two nodes are connected by an edge the parameter estimates for the corresponding components of the phenotype are fused together via the ℓ_1 norm. For more adaptability, the terms inside the fusion penalty are weighted by the amount of correlation between the components of the phenotype, which results in the *graph-weighted fused lasso* (G_wFlasso) defined as

$$\widehat{\boldsymbol{B}}_{\text{G}_\text{w}}(\lambda, \mu) = \underset{\boldsymbol{B}}{\arg\min} \|\boldsymbol{Y} - \boldsymbol{XB}\|_2^2 + \lambda\|\boldsymbol{B}\|_1 + \mu \sum_{k,l=1}^{p} w(r_{kl})\|\boldsymbol{\beta}_{\bullet k} - sgn(r_{kl})\boldsymbol{\beta}_{\bullet l}\|_1. \tag{A2}$$

The tuning parameters λ and μ determine the amount of penalization by each penalty, while $\boldsymbol{\beta}_{\bullet k}, \boldsymbol{\beta}_{\bullet l}$ are the kth and lth columns of $\boldsymbol{B}$, respectively, and r_{kl} is the Pearson correlation coefficient of the response vectors $\boldsymbol{Y}_k$ and $\boldsymbol{Y}_l$ (columns of $\boldsymbol{Y}$) and $w(r)$ is a weight function, i.e. a non-negative function on $(-1, 1)$. Additionally, Kim *et al.* [12] require $w(r)$ to be equal to 0 on $(-\rho, \rho)$ where $\rho \in (0, 1)$ is a suitably chosen cut-off value so that the pairs of phenotypes with correlation coefficient below ρ (in absolute value) do not enter the fusion penalty.